



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
|-----------------|-------------|----------------------|---------------------|------------------|

10/741,516

12/19/2003

Thomas O'Rourke

920673-95339

4435

23644 7590 04/10/2007  
BARNES & THORNBURG LLP  
P.O. BOX 2786  
CHICAGO, IL 60690-2786

EXAMINER

HICKS, MICHAEL J

ART UNIT

PAPER NUMBER

2165

| SHORTENED STATUTORY PERIOD OF RESPONSE | MAIL DATE | DELIVERY MODE |
|--|-----------|---------------|
|--|-----------|---------------|

3 MONTHS

04/10/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

**Office Action Summary**

Application No.

10/741,516

Applicant(s)

O'ROURKE ET AL.

Examiner

Michael J. Hicks

Art Unit

2165

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 31 January 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 19 December 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## DETAILED ACTION

1. Claims 1-24 Pending.

### *Response to Arguments*

2. Applicant's arguments, see response, filed 1/31/2007, with respect to the rejection(s) of claim(s) 1-24 under USC 102 have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Stal (Web Services: Beyond Component-Based Computing", Communications of the ACM, Vol. 45, No. 10, Pages 71-76, October 2002) and Bou-Ghannam et al. (U.S. Pre Grant Publication Number 2005/0102350 and referred to hereinafter as Bou).

### *Claim Rejections - 35 USC § 102*

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-5, 7-17, and 19-24 rejected under 35 U.S.C. 102(b) as being anticipated by Stal.

As per Claim 1, 11, and 12, Stal discloses a computer program product, server (e.g. system), and method of providing a client computer with remote access to an application controlled by a server across a data network (i.e. The Stal reference deals with Web Services which are remote applications on a server, which are controlled by a remote client across a network (generally the internet).) **without maintaining a dedicated communications channel between the client and the server** (i.e. *"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services...Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates that the Web services are accessed through a standard web browser interface. Note that according to Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4), **comprising the steps of: a) providing a network addressable server-side service which executes one or more predefined procedures to control said application in response to predefined application control commands received by the server over the network** (i.e. Figure 2 clearly indicates that the client sends calls/methods/application control commands over the network to the server, which executes predefined procedures in response to those commands and sends back a result to the client.); **b) providing a definition of said application control commands, said definition being accessible over the data network by a client to enable the client to provide an interface which formulates said predefined commands and addresses said commands to the server-side service in response to inputs to the client** (i.e. *"Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the*

Art Unit: 2165

specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service." The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1), c) providing instructions accessible over the data network by the client which define a network addressable client-side service which executes one or more predefined procedures to generate notifications on the client in response to predefined notification commands received by the client over the network (i.e. "• Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group

Art Unit: 2165

*messages to abstract operations. • Bindings map Porttypes to concrete protocols."* The preceding text excerpt along with Figure 2 clearly indicates that the WSDL definition, which is accessible over the data network via the directory, provides definitions of appropriate responses to data received at the client. Note that in order to inform the user of the changes notification request must be generated on the client and reflected through a user interface to the user.) (Page 74, Column 2, Paragraph 1); and d) providing on the server an interface between the application and the data network which is effective to issue one or more of said notification generation commands in response to notifications of events received from the application (i.e. Figure 2 clearly indicates that a server proxy may exist (e.g. an interface between the server application and the data network) which passes results (e.g. notification generation commands) in response to receiving notifications from the server.); whereby the server-side service provides a network addressable control service to enable the client to control said application on the server (i.e. The Stal reference deals with Web Services which are remote applications on a server, which are controlled by a remote client across a network (generally the internet).), and the client-side service provides a network addressable notification service to enable the server to notify the client of events occurring in the application (i.e. Figure 2 clearly indicates that the server notifies the client of the results of requests through the data network.).

As per Claims 2 and 14, Stal discloses said server-side service and said client-side service are each provided as web services between a provider and a remote consumer, the consumer of the server-side service being the client interface (i.e. Figure 2 clearly indicates that the client is the consumer of the server side service in that data from the client to the server is sent in order to trigger operations (e.g. processes) in the server.) and the consumer of the

Art Unit: 2165

client-side service being the server interface (i.e. Figure 2 clearly indicates that the server is the consumer of the client side service in that data from the server to the client is sent in order to trigger operations (e.g. event/result notifications) in the client.).

As per Claims 3 and 15, Stal discloses the definition of said application control commands is a web services description language (WSDL) file provided on the server (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation-neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports." The preceding text excerpt clearly indicates that WSDL is used to describe the definition of the web services application control commands. Note that Step # indicates that the WSDL definition files are stored on the server.) (Page 74, Steps 2 and 3).

As per Claims 4 and 16, Stal discloses said WSDL file includes said instructions which define said client-side web service (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation-neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and

*serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents:*

- Types are used as core elements to build messages (XML Schema Notation).*
- Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages.*
- Porttypes group messages to abstract operations.*
- Bindings map Porttypes to concrete protocols.*
- Ports denote the concrete communication addresses of services.*
- A Service comprises a collection of ports."*

The preceding text excerpt clearly indicates that WSDL is used to describe the definition of the web services application control commands. Note that Step # indicates that the WSDL definition files are stored on the server.) (Page 74; Steps 2 and 3).

As per Claims 5 and 17, Stal discloses said WSDL file includes instructions for the client to access said instructions which define said client-side web service over the network (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents:

- Types are used as core elements to build messages (XML Schema Notation).*
- Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages.*
- Porttypes group messages to abstract operations.*
- Bindings map Porttypes to concrete protocols.*
- Ports denote the concrete communication addresses of services.*
- A Service comprises a collection of ports."*

The preceding text excerpt clearly indicates that WSDL is used to describe the definition of the web services application control commands. Note that Step # indicates that the WSDL



definition files are stored on the server.) (Page 74, Steps 2 and 3).

As per Claims 7 and 19, Stal discloses said data network is a packet switching network employing the transport communication protocol/internet protocol (TCP/IP) method of addressing packets from the client to the server and vice versa (i.e. *"Basically, client and servant programmers prefer a method invocation paradigm that fits nicely into the object models of their favorite programming languages. However, ultimately every communication paradigm is layered on top of a low-level transport protocol such as TCP/IP. As a consequence, each remote communication must be handled by message passing. Thus, in the guts of every implementation requests, responses, and faults are represented as first-class objects upon transmission over the wire."*) The preceding text excerpt clearly indicates that the network communications are performed using TCP/IP.) (Page 72, Column 2, Paragraph 3).

As per Claims 8 and 20, Stal discloses the data network is selected from a local area network, a wide area network, and the Internet (i.e. *"Web-based, front-end layers must provide both Web server functionality and Web service provisioning."*) The preceding text excerpt clearly indicates that the services use Web (e.g. Internet) interfaces, thus the network is the Internet.) (Page 74, Column 2, Paragraph 4).

As per Claim 9, Stal discloses said server is a web server running said application (i.e. Figure 2 clearly indicates that the server may be a web server running the application.).

As per Claim 10, Stal discloses said server is a web server and said application runs on a computer in communication with said server (i.e. Figure 3 clearly indicates that the

Art Unit: 2165

server may be a web server, and the application may be running on a computer (e.g. another web server) in communication with the server.).

As per Claims 13, 21, and 22, Stal discloses a computer program product, client (e.g. system), and method of remotely controlling an application from a client computer across a data network, said application being under the local control of a server on the data network (i.e. The Stal reference deals with Web Services which are remote applications on a server, which are controlled by a remote client across a network (generally the internet).), **without maintaining a dedicated communications channel between the client and the server** (i.e.

*"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services...Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates that the Web services are accessed through a standard web browser interface. Note that according to

Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4), comprising the steps of: a) receiving from the server a definition of

application control commands which cause a network addressable server-side service to execute one or more predefined procedures to control said application (i.e. "Step 2:

*Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core*

elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service." The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); b) providing on the client an interface based on said definition which formulates said predefined commands and addresses said commands to the server-side service in response to inputs to the client (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define

Art Unit: 2165

*packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions: The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service."* The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); c) receiving

instructions over the data network which define a network addressable client-side service which executes one or more predefined procedures to generate notifications on the client in response to predefined notification commands received by the client over the network (i.e. "*• Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols.*"

The preceding text excerpt along with Figure 2 clearly indicates that the WSDL definition, which is accessible over the data network via the directory, provides definitions of appropriate responses to data received at the client. Note that in order to inform the user of the changes notification request must be generated on the client and reflected through a user interface to the user.) (Page 74, Column 2, Paragraph 1); and d) providing on the client said network

addressable client-side service which monitors for receipt of said notification generation commands and which generates said notifications on the client in response thereto (i.e.

*"• Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols."* The preceding text excerpt along with Figure 2 clearly indicates that the WSDL definition, which is accessible over the data network via the directory, provides definitions of appropriate responses to data received at the client. Note that in order to inform the user of the changes notification request must be generated on the client and reflected through a user interface to the user.)

(Page 74, Column 2, Paragraph 1); whereby the server-side service provides a network addressable control service to enable the client to control said application on the server (i.e. The Stal reference deals with Web Services which are remote applications on a server, which are controlled by a remote client across a network (generally the internet).), and the client-side service provides a network addressable notification service to enable the server to notify the client of events occurring in the application (i.e. Figure 2 clearly indicates that the server notifies the client of the results of requests (e.g. events occurring in the application) through the data network, thus a service must be in place to enable the client to receive the notifications.).

As per Claim 23, Stal discloses a system comprising a client and a server connected across a data network, the client and the server each being provided with a storage area for storing instructions to allow asynchronous interaction between the client and server, and the server having control of an application (i.e. *"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services...Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates that the

Art Unit: 2165

Web services are accessed through a standard web browser interface. Note that according to Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4), wherein: (I) the storage area of the server stores instructions which when executed are effective to cause the server to: a) provide a network addressable server-side service which executes one or more predefined procedures to control said application in response to predefined application control commands received by the server over the network (i.e. Figure 2 clearly indicates that the client sends calls/methods/application control commands over the network to the server, which executes predefined procedures in response to those commands and sends back a result to the client.); b) provide a definition of said application control commands, said definition being accessible over the data network by the client to enable the client to provide an interface which formulates said predefined commands and addresses said commands to the server-side service in response to inputs to the client (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must

*find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service."* The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); c) provide instructions accessible over the data network by the client which define a network addressable client-side service which executes one or more predefined procedures to generate notifications on the client in response to predefined notification commands received by the client over the network (i.e. "*• Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols.*") The preceding text excerpt along with Figure 2 clearly indicates that the WSDL definition, which is accessible over the data network via the directory, provides definitions of appropriate responses to data received at the client. Note that in order to inform the user of the changes notification request must be generated on the client and reflected through a user interface to the user.) (Page 74, Column 2, Paragraph 1); and d) provide on the server an interface between the application and the data network which is effective to issue one or more of said notification generation commands in response to notifications of events received from the application (i.e. Figure 2 clearly indicates that a server

Art Unit: 2165

proxy may exist (e.g. an interface between the server application and the data network) which passes results (e.g. notification generation commands) in response to receiving notifications from the server.) (i.e., Figure 2 clearly indicates that a server proxy may exist (e.g. an interface between the server application and the data network) which passes results (e.g. notification generation commands) in response to receiving notifications from the server.); and (II) the storage area of the client stores

instructions which when executed are effective to cause the client to: a) receive from the server across the network said definition of application control commands (i.e. "Step 2:

*Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3:*

*Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see*

*[www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy,*



Art Unit: 2165

and invokes the Web service." The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); b) provide on the client an interface based on said definition which formulates said predefined commands and addresses said commands to the server-side service in response to inputs to the client (i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation- neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations. UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service." The preceding text excerpt clearly indicates that a service definition

Art Unit: 2165

language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input. Also this definition is available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); c) receive over the

**data network said instructions which define a network addressable client-side service**

(i.e. "Step 2: Implementing an Interface Definition Language. In the next implementation step, an interface definition language is introduced to express structural and deployment information in an implementation-neutral fashion. From the specifications written in this language, tools will automatically generate the implementations of client-side and serverside proxies. Again, XML represents the appropriate means to define such a data representation language. As a result WSDL (Web Services Description Language)—see [www.w3c.org/TR/wsdl](http://www.w3c.org/TR/wsdl)—was created providing the following constituents: • Types are used as core elements to build messages (XML Schema Notation). • Messages define packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete protocols. • Ports denote the concrete communication addresses of services. • A Service comprises a collection of ports. Step 3: Implementing a Service Directory. Before a client can access a service, it must find the service. For this purpose, a central broker must be available that allows implementers to register their services as well as clients to locate these services. Again, XML denotes the core technology to store and retrieve service registrations: UDDI (Universal Discovery, Description, and Integration)—see [www3.ibm.com/services/uddi/standard.html](http://www3.ibm.com/services/uddi/standard.html)—provides all functionality expected from a service broker. In UDDI, servers use the Publishers API to register services as well as additional business information with the global repository (see Figure 3). Clients access the Inquiry API to browse the repository and retrieve service descriptions. SOAP is used as communication protocol in all interactions. The client obtains the WSDL description from the UDDI repository both dynamically or statically, generates a client-side proxy, and invokes the Web service." The preceding text excerpt clearly indicates that a service definition language (WSDL) is used to create a description of the application control commands and the interface, including responses to user input, all of which describe the client side service. Also this definition is

Art Unit: 2165

available to be sent over the network from the server to the client.) (Page 74, Column 1, Paragraph 3, Column 2, Paragraph 1); and d) provide on the client said network addressable client-side service which monitors for receipt of said notification generation commands and which generates said notifications on the computer in response thereto (i.e. "• Messages define

*packages exchanged within a single message transfer. Requests and responses represent separate messages. • Porttypes group messages to abstract operations. • Bindings map Porttypes to concrete*

*protocols."* The preceding text excerpt along with Figure 2 clearly indicates that the WSDL definition, which is accessible over the data network via the directory, provides definitions of appropriate responses to data received at the client. Note that in order to inform the user of the changes notification request must be generated on the client and reflected through a user interface to the user.) (Page 74, Column 2, Paragraph 1); whereby the server-side service provides a network addressable control

service to enable the client to control said application on the computer (i.e. The Stal reference deals with Web Services which are remote applications on a server, which are controlled by a remote client across a network (generally the internet).), and the client-side service provides a network addressable notification service to enable the server to notify the client of events occurring in the application (i.e. Figure 2 clearly indicates that the server notifies the client of the results of requests (e.g. events occurring in the application) through the data network, thus a service must be in place to enable the client to receive the notifications.), without maintaining a

dedicated communications channel between the remote client and the computer (i.e.

*"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services...Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates

that the Web services are accessed through a standard web browser interface. Note that according to Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without

Art Unit: 2165

maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4).

**As per Claim 24, Stal discloses a method of providing an asynchronous interaction between a client and a server** (i.e. *"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services... Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates that the Web services are accessed through a standard web browser interface. Note that according to Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4), **comprising the steps of providing Web Services on the server to be consumed by the client** (i.e. Figure 2 clearly indicates that the client is the consumer of the server side service in that data from the client to the server is sent in order to trigger operations (e.g. processes) in the server.) **and providing Web Services on the client to be consumed by the server** (i.e. Figure 2 clearly indicates that the server is the consumer of the client side service in that data from the server to the client is sent in order to trigger operations (e.g. event/result notifications) in the client.), **whereby each of the client and server acts as both a Web Services provider and a Web Services consumer, such that when acting as a consumer each can notify the other of events asynchronously by invoking a Web Services command** (i.e. Figure 2 clearly indicates clearly indicates that the client notifies the server of events using request/method/call messages and the server notifies the client of events using result/notification generation messages. As the commands used to invoke these messages are associated with a Web Service, they may be condiered to be Web service

Art Unit: 2165

commands.), and wherein said asynchronous interaction is provided without maintaining a dedicated communications channel (i.e. *"The basic idea is to build Broker-based middleware using Internet protocols such as HTTP and XML as a data marshaling solution. This is the essence of Web services...Web-based, front-end layers must provide both Web server functionality and Web service provisioning."* The preceding text excerpt clearly indicates that the Web services are accessed through a standard web browser interface. Note that according to Applicants specification (Page 6, Lines 3-22) indicate that a standard browser interface utilizing a standard internet connection fulfills the limitations of providing asynchronous communications without maintaining a dedicated communications channel.) (Page 71, Column 1, Paragraph 1; Page 74, Column 2, Paragraph 4).

### ***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. Claims 6 and 18 rejected under 35 U.S.C. 103(a) as being unpatentable over Stal in view of Bou.

As per Claims 6 and 18, Stal fails to disclose said application is a computer telephone integration (CTI) application in communication with a telephony network, whereby the interface on the client enables a user of the client to control a device on said telephony network and the interface on the server enables the CTI application to

Art Unit: 2165

issue notifications of telephony network events or state changes relating to the device to the user of the client.

Bou discloses said application is a computer telephone integration (CTI) application in communication with a telephony network (i.e. *"In this example, Web service 225 can return a URL of another Web service that can dial a telephone number. Web service 230 can return a telephone number if provided a name. Web service 235 can dial a telephone number."* The preceding text excerpt clearly indicates that the web service is in communication with a telephony network (e.g. can access a phone in the network.) (Page 3, Paragraph 33), whereby the interface on the client enables a user of the client to control a device on said telephony network (i.e. *"In this example, Web service 225 can return a URL of another Web service that can dial a telephone number. Web service 230 can return a telephone number if provided a name. Web service 235 can dial a telephone number."* The preceding text excerpt clearly indicates that the device in the telephony network can be controlled via the Web Service.) (Page 3, Paragraph 33) and the interface on the server enables the CTI application to issue notifications of telephony network events or state changes relating to the device to the user of the client (i.e. *"In this example, Web service 225 can return a URL of another Web service that can dial a telephone number. Web service 230 can return a telephone number if provided a name. Web service 235 can dial a telephone number...After the phone number is dialed, watcher 220 can end or, alternatively, can modify the pattern object 205 by adding an identifier or some other reference prior to terminating execution. "* The preceding text excerpt clearly indicates that a notification of when the call is connected.) (Page 3, Paragraphs 33 and 37).

It would have been obvious to one skilled in the art at the time of Applicants invention to modify the teachings of Stal with the teachings of Bou to include said application is a computer telephone integration (CTI) application in communication with a telephony network, whereby the interface on the client enables a user of the client to

Art Unit: 2165

control a device on said telephony network and the interface on the server enables the CTI application to issue notifications of telephony network events or state changes relating to the device to the user of the client with the motivation of processing requests for a plurality of Web Services (Bou, Abstract).

### ***Points of Contact***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Hicks whose telephone number is (571) 272-2670. The examiner can normally be reached on Monday - Friday 8:30a - 5:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on (571) 272-4146. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Application/Control Number: 10/741,516  
Art Unit: 2165

Page 23

Michael J Hicks  
Art Unit 2165  
Phone: (571) 272-2670  
Fax: (571) 273-2670



JEFFREY GAFFIN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100